SP88－17

拡張連続ＤＰ法による連続音声認識装置とその応用

A Continuous Speech Recognizer Using Augmented Continuous DP matching and Its Application

小川　一朗＊　　中川　聖一＊
Ichiro OGAWA＊　　Seiich NAKAGAWA＊

＊豊橋技術科学大学情報工学系
＊Department of Information & Computer Sciences, Toyohashi University of Technology

あらまし　ユーザと機械との好ましい音声による対話法を研究したり、音声認識装置をユーザが容易に利用できるための使用法等を研究するためには、オンライン実時間動作可能な音声認識装置の開発が必要不可欠である。本論文では、汎用ＤＳＰ（ＩｍＰＰ）及び拡張連続ＤＰ法を用いた連続音声認識装置、ユーザがこの認識装置を容易に応用できるように認識対象タスクと認識結果の意味抽出法をユーザが容易に定義できるユーザインターフェース、さらに装置の日本語ＬＯＧＯへの適用について述べる。

Abstruct　This paper describes a continuous speech recognizer, which uses DSP(ImPP) and the Augmented Continuous DP matching algorithm. Additionally, to improve its flexibility or user-interface, the expression method of automaton for a given task and the extraction of any necessary information for the application from the recognition results were built in the equipment. And we apply the recognizer to Japanease-LOGO.

1．はじめに

音声言語は、人間にとって思考する道具であり、コミュニケーションの道具である。この様な観点から機械と人間とのコミュニケーションを考えた場合、音声言語をコミュニケーションの手段とすることは自然な発想である。それ故、発話音声の機械による認識は古くからの研究課題であった。

音声認識の研究は、特定話者の小数語いの単語音声認識から連続（単語）音声認識や不特定話者の大語い単語音声認識へと進んでおり、連続音声認識用の専用装置も開発されている。筆者らは、パターンマッチング法に基づく連続音声認識法に対して、極めて効率のよい拡張連続ＤＰ法を考案した[1]。本アルゴリズムは、迫江の考案した２段ＤＰ法[2]よりも１０倍以上高速であり、ほぼ同じ認識能力を持つ。

ＤＰマッチングによる連続単語認識法は、計算量が多く汎用のＤＳＰやマイクロプロセッサで実現された例は極めて少なく[10]、通常何等かの計算量削減化法を使用しなければならない[3]。

基礎研究段階においても実時間に近い速度で動作するオンライン音声認識装置を開発することは重要である。その主な点を列挙すれば、

①ユーザと機械（音声認識装置）との音声による対話法を研究するためにはオンライン実時間によるシミュレーションが必要不可欠である。

②ユーザが容易に音声認識装置を利用出来るための使用環境、タスクに依存した初期設定法等を研究開発するには、認識装置の存在が前提となる。

③装置のＬＳＩ化の為の種々の問題点や改善策等の知見を得る。

④認識法の評価が即座に実行でき、研究のスピードアップ化をもたらす。

本研究では①～③の目的の為に行ったもので、①の目的の為に計算量の少ない拡張連続ＤＰ法を用いることにより、汎用のＤＳＰによって連続音声認識装置を実現した。また②の目的の為に、ユーザが希望するタスクを容易に設定できるためのオートマトンによるタスク定義法と認識結果をユーザが容易に利用できるためのユーザ定義による意味抽出法を開発した。この様な研究は音声認識装置の実用化・普及にとって極めて重要であるが今まで研究された例は極めて少ない[4],[5]。

本論文ではまず２章で基本となる拡張連続ＤＰ法を簡単に説明し、３章で汎用のＤＳＰを用いた連続音声認識装置について述べ、４章でユーザフレンドリな音声認識装置の為のタスク定義法と意味抽出法、５章で装置の評価、６章で実際のアプリケーション例として日本語ＬＯＧＯへの適用について述べる。

2. 拡張連続ＤＰ法

はじめに、本装置で採用した拡張連続ＤＰ法について説明する。詳細は文献[1],[7]を参照されたい。

2.1 連続単語音声認識問題の定式化と解法[1],[2],[6],[7]

2.1.1 記号の定義

n ：単語番号．（各単語名に割り当てられた自然数）

a_i ：入力音声の第i区間（フレームと呼ぶ）の特徴パラメータのベクトル表現．

b^n_j ：単語nの標準パターンの第jフレームの特徴パラメータのベクトル表現．

I ：入力パターン長（フレーム数），

J^n ：単語nの標準パターン長（フレーム数），

q ：状態番号，

q_0 ：初期状態番号，

$d^n(i,j)$ ：単語nの第jフレームの（b^n_j）と入力の第iフレーム（a_i）とのフレーム間距離．

$D(q,i)$ ：入力パターンのフレーム区間（$a_1 a_2 \cdots a_i$）と状態qに達する単語列の標準パターンとの最小累積照合距離．

$N(q,i)$ ：$D(q,i)$に対応する単語列の最後尾単語名．

$B(q,i)$ ：$N(q,i)$の入力パターン系列上における存在開始フレーム番号－１（バックポインタ），

$Q(q,i)$ ：$D(q,i)$に対応する単語列の状態qの直前の状態，$\delta(Q(q,i),N(q,i))=q$，

2.1.2 定式化と解法

入力パターンの部分パターン$a_{m+1}, a_{m+2}, \cdots, a_i$と標準パターン$b^n_1, b^n_2, \cdots, b^n_{J^n}$とのＤＰマッチング累積距離を$D^n(m+1:i)$と表すと

$$D(q_0,0)=0,\quad B(q_0,0)=0$$
$$D(q,i)=\min_{n,m,q_k}\{D(q_k,m)+D^n(m+1:i)\} \qquad (1)$$

ただし$\{q_k, n \mid q=\delta(q_k,n)\}$，$\{0 \le k \le |S|-1\}$

となる。(1)式は状態q_kから単語nを生成して状態qに達することを意味する。

(1)式を満たすn,m,q_kを$\hat{n}$,$\hat{m}$,$\hat{q}_k$とし、$N(q,i)=\hat{n}$，$B(q,i)=\hat{m}$，$Q(q,i)=\hat{q}_k$とする。

(1)式を$1 \le i \le I$で繰り返した後、認識結果は次のようにして求める（判定処理）。

①最適最終状態　$\hat{q}=\arg\min_{q_r \in \{最終状態\}} D(q_r,I)$

②初期条件　$i=I$，$q=\hat{q}$

③認識結果の出力　$\hat{n}=N(q,i)$

状態遷移　$\hat{q}=Q(q,i)$

時刻遷移　$\hat{i}=B(m,q)$

④$i \neq 0$ならば　$q=\hat{q}$，$i=\hat{i}$として③へ

こうして、最適な単語列の単語名と照合位置が、最後尾単語から先頭単語まで順次求まる。

2.2 拡張連続ＤＰ法による連続単語音声認識アルゴリズム

2.2.1 記号の定義

$\bar{D}^n(i,j)$ ：入力パターンのフレーム区間（$a_m a_{m-1} \cdots a_i$）と標準パターンnのフレーム区間（$b^n_1 b^n_2 \cdots b^n_j$）との最小累積距離のmについての最小値。ここで、ＤＰパスは標準パターン側の軸を基本軸とする。

$\bar{B}^n(i,j)$ ：$\bar{D}^n(i,j)$に対応する入力パターンの始点位置。

$\bar{D}^n(i,j)$は次のように定式化できる。

$$\bar{D}^n(i,j)=\min_{v \in V}\sum_{k=1}^{j} d^n(v(k),k) \qquad (2)$$

ここで、Vは標準パターンと入力パターンを対応づけする時間関数群で、例えば$0 \le v(k)-v(k-1) \le 2, 1 \le v(1) \le i, v(j)=i, v(1)$の初期条件は、入力パターンに関して始端点フリーを実現するためのものである。

2.2.2 オートマトン制御拡張連続ＤＰ法による連続単語音声認識アルゴリズム

拡張連続ＤＰ法は、2.1.1で定義した$D^n(m+1:i)$をword spotting法によって$\bar{D}^n(i,J^n)$，$\bar{B}^n(i,J^n)$から近似的に求める。処理の流れは、フレーム同期で進む実時間認識向きアルゴリズムになっている。

＜アルゴリズム＞

①初期条件

$D(q_0,0)=0$，$B(q_0,0)=0$

$D^n(-1,j)=D^n(0,j)=\infty$　for

$n=1,2,\cdots,N$；$j=1,2,\cdots,J^n$

②$i=1,2,\cdots,I$について③～⑨を実行，

③$n=1,2,\cdots,N$について④～⑦を実行。

④始端点フリーのための初期化

$\bar{D}^n(i,1)=d^n(i,1)$，$\bar{B}^n(i,1)=i$

⑤$j=2,3,\cdots,J^n$について⑥を実行。

⑥$i=\arg\min_{i-2 \le i' \le i} \bar{D}^n(i',j-1)$

$\bar{D}^n(i,j)=\bar{D}^n(i,j-1)+d^n(i,j)$

$\bar{B}^n(i,j)=\bar{B}^n(i,j-1)$

⑦$r=-r_1,-r_1+1,\cdots,-1,0,1,\cdots,r_2$について

$m+1=\bar{B}^n(i,J^n)$

$D^n(m+r+1:i)=\bar{D}^n(i,J^n)\times(j-m+r)/J^n$

⑧$q=q_1,q_2,\cdots,q_{|S|-1}$について⑨を実行。

⑨$m=\bar{B}^n(i,J^n)-r_1-1,\cdots,\bar{B}^n(i,J^n)+r_2-1$について、且つ、$n=1,2,\cdots,N$について

$\hat{m},\hat{n},\hat{q}=\arg\min_{m,n,q'}\{D(q',m)+D^n(m+1:i)\}$

ただし、$\{n,q' \mid q=\delta(q',n)\}$

$D(q,i)=D(\hat{q}',\hat{m})+D^{\hat{n}}(\hat{m}+1:i)$

$N(q,i)=\hat{n}$

$B(q,i)=\hat{m}$

$Q(q,i)=\hat{q}'$

⑩判定処理（2.1.2参照），

④～⑥は、各単語パターン列の並列処理可能な計算量の多い単純な繰り返し計算である。

3. 連続音声認識装置

3.1 第１版の連続音声認識装置[8]の問題点

ディジタル信号処理プロセッサμPD7720を用いた第１版の連続音声認識装置を試作した[8]が、処理速度、登録単語数などに関して不十分であった。この装置は128kバイトの外部メモリボード、音声分析ボード、ＤＰマッチングボード、及びホストコンピュータからなり、単語標準パターン約５０個が登録できるコンパクトな装置であったが次の問題点が明かとなった。

①音声分析は、μPD7720を２個用いたデジタルフィ

ルタによって行っていたがソフトウェアによる分析（FFT分析、LPC分析）に比べて精度があまり良くなく、認識率の低下をもたらした[8]。

②計算量のもっとも多い局所距離計算と累積距離計算（2.2.2アルゴリズム④～⑤の部分）はμPD7720を1個で、単語連結等の後処理（⑦～⑨の部分）はホストコンピュータ（PC-9801）で行い、これらを並列動作で実行したが、語い数が30を越えれば後処理の方が処理時間がかかることが明かとなった。

そこで、第2版の連続音声認識装置として、拡張連続DP法のアルゴリズムの特徴を生かした並列処理による高速化と登録語い数の増大を図った。

3.2 第2版の連続音声認識装置

3.2.1 装置の特徴

本装置のハードウェア的な特徴を以下に述べる。

本装置の第一の特徴は、市販ハードウェアを用いて認識システムを構築している為、高い再現性と保守性を持っている点である。

第二の特徴は、パターンマッチング部にデータ・フロー方式のデジタル信号処理プロセッサImPP(μPD7281)を用いている点である。ImPPは本来、イメージ処理を高速に行うことを目的に設計されたＬＳＩであるが、

1)可変パイプライン方式による高速データ処理
2)高速演算機能
3)マルチプロセッサ指向
4)高速データ入出力機能

等の特徴を持つので、ホストプロセッサと並列動作可能な汎用高速データ処理プロセッサと言える。ここで一般の信号処理用汎用ＤＳＰ（例、μPD7720）と比較した場合アーキテクチャ的に大きく異なるのは、データ・フロー方式を採用している点である。この特徴によりマルチプロセッサ時のハードウェア構成が非常にシンプルである[12]。またマルチプロセッサ環境に於けるソフトウェア開発の見地から述べるなら、データ・フロー方式の採用により、プロセッサ間のコミュニケーションの細かいタイミングをユーザが考慮しなくてもよいため、各プロセッサへの機能分散が容易に行える。

3.2.2 装置の概要

a) 装置の仕様

第2版の装置としてマルチプロセッサ構成の容易なImPP（μPD7281）を用いてＤＰマッチング部をＤＳＰ化した認識装置を開発した。TiP-PCボードⅡ（日本電機の試作品）という、ホストコンピュータPC-9801の拡張スロットに差し込んで使用できるものを使用した。表1に仕様を示す。

認識語い数は、外部メモリの容量の制限により1単語平均50フレーム、特徴パラメータは14ch／フレームのデータとして約140単語である。分析方法はFFT分析（14チャンネルにデータ圧縮）またはFFTケプストラム分析（14次）である。

b) 音声認識装置の構成

図1に装置の構成を示す。ホストマシンのPC-9801Eで各部の制御を行う。音声はカットオフ周波数4.4KHzのローパスフィルタで高域成分を除去した後、

表1　装置の仕様（第2版）

方式	拡張連続ＤＰ法
発声法	連続発声可
認識語い数	最大１４０単語
最大状態数	５０状態
分析方法	１４ｃｈＦＦＴスペクトル １４次　ＦＦＴケプストラム
分析周期	１０ｍｓ
認識時間	１０数字の連続音声では実時間
認識精度	特定話者の連続音声中の 数字認識率　約９５％～９８％
その他	オートマトンによる構文制御 認識結果からの意味抽出 いずれもユーザが簡便に定義可

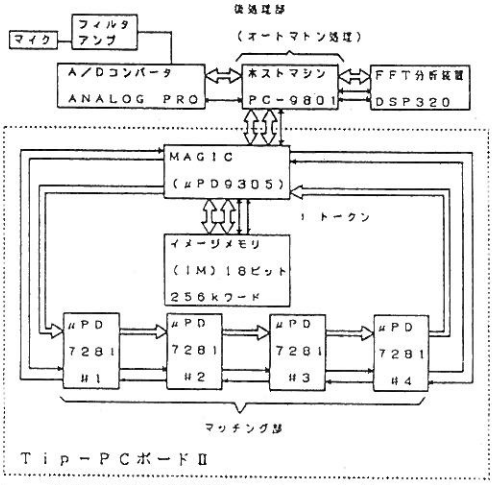

図1　連続音声認識装置のブロックダイアグラム

10KHzでサンプリングされる。この256点にハミング窓をかけてフーリエ変換を行い128本のスペクトルを14個に圧縮する。フレーム周期は10msである。なお音声認識に有効だと言われているケプストラム係数を本ボードで求めることが出来る。音声入力部のＡ／Ｄコンバータは“カノープス製ANALOG PRO（分解能：12bit，変換時間：12μs）”を、音声分析部はFFT分析ボード（中部電機製　ＤＳＰ-320 256点実数FFT：6.1ms，256点逆FFT：12.1ms）を使用した。認識の主な計算は、TiP-PCボードⅡで行われる。TiP-PCボードⅡは、PC-9801の拡張スロットに挿入して使用でき、データ・フローアーキテクチャを採用したImPP（μPD7281）４個とMAGIC（Memory Access and General bus-Interface Chip μPD9305）１個でリングを構成し、IM(Image memory) 256KbitのDRAM18個からなる。４個のImPPをマッチング部に用いる。尚、後処理部はホストコンピュータで行う。後処理部はビームサーチ法の採用によって高速化できる。

c) 処理の流れ

予め登録しておいた標準パターンは、ホストコンピュータのディスクからTiP-PCボードⅡ上のIMへ転送される。発声された音声は、10kHz でサンプリン

グされホストマシンへ入力される。これをFFT分析ボードへ送り、パラメータに変換された後再びホストマシンに返される。分析結果は、ディスプレイ上で、エネルギー、ソナグラムとして表示され適切な入力であったか確認できる。

ホストマシンは分析された結果をIMへ転送し、認識を開始する準備を行う。すなわち、初期化を行いImPPに起動トークンを出力する。ImPPでは、パターンマッチング、バックポインタの計算が行われる。その結果をホストマシンに返し後処理を行った後、最終結果を出力し、必要ならばその認識結果から必要なパラメータ（意味表現）を取り出す。

d) ホストマシン

ホストマシンは、標準・入力パターンのメモリボードへの転送、各部の制御及びマッチングの後処理を行っている。

d-1) 音声分析時

A/D コンバータからの変換終了を待って、音声の生データを受け取る。現在は、一旦音声の生データを入力し終えた後にFFT分析ボードに対して分析のコマンドが送られている。

d-2) 認識時

パラメータに変換された音声をIMへ転送し、ImPPに対して起動トークンを出力する。

3.2.2 マッチング部の並列処理

汎用のＤＳＰを用いてＤＰマッチングを高速に実行するためには処理を並列化せざるを得ない。

a) 処理の分割法の検討

パターン照合部の一連の処理を複数のプロセッサで行う場合、その処理の分割方法（分散方法）が問題になる。その方法には、一般に大きく分けて次の２通りの方法が考えられる。

①データ処理領域の分割（負荷分散）
　1)入力パターン領域の分割
　2)参照パターン領域の分割
②処理内容の分散（機能分散）

①の方法は、入力パターン領域では幾つかの区間に、参照パターン領域では幾つかの参照パターン群に分割し、同じプログラムがロードされている複数個のμPD7281を用いて、それぞれの区間内の処理を並列に行う方法である。拡張連続ＤＰ法はいずれの領域でも分割可能であるが、2)の方法の場合には、参照単語長の違いによるプロセッサ間の負荷のアンバランスが生じる可能性があることと、単語群ごとのバンク切り替えが必要になるので、1)の方法を採用した。また、②の方法は、①の場合とは違って、処理対象が分割しにくい場合や、プログラムが１個のμPD7281に入りきらない場合に使用される方法である。１個分のμPD7281のプログラム領域では一連の処理プログラムが全て格納できない理由によりこの分割法も取り入れた。以上のような問題や制約を考慮し、また、拡張性のある装置を目標に図２に示すような処理分担法を採用した。図２に示すように四つのImPPで機能分散と負荷分散を行っており、ImPP1とImPP3、ImPP2とImPP4との各々のペアで負荷分散を、ペアの内部で機能分散を行っている。

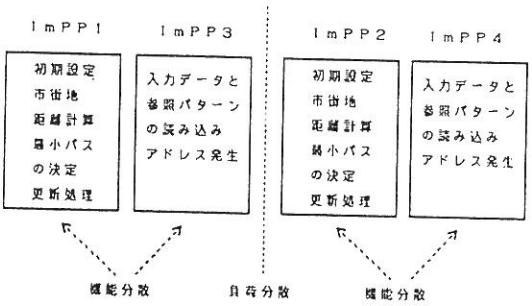

図２　処理プログラムの分割

4. ユーザフレンドリなインタフェース

4.1 タスクのユーザ定義

－構文情報を決定するオートマトンの記述－

本装置で扱う入力音声の構文情報は有限オートマトンで表現できるものに限っている。一般の自然言語文は有限オートマトンでは表現できなくなるが、本装置のように語い数が140単語程度のタスクについては有限オートマトンに限っても不自然ではなく、応用分野もかなり存在すると思われる。

実際のユーザが音声認識装置を応用する場合、認識する対象文（通常タスクと呼ぶ）の構文の記述入力が容易にできなければならない。その記述をプログラムに組み込む必要があるならば、一般のユーザは使えなくなる。そこで、ユーザによるタスクの記述および認識時のタスク選択を簡便にするために、音声認識装置とユーザとのインタフェースを設計した。このユーザインタフェースを用いたときのタスク記述から認識までにユーザの行う処理を以下に示す。

1)ユーザによるタスク記述。

ユーザはオートマトン表現したタスクをテキスト形式のタスクファイルに記述する事によってタスクを表現する。タスクの表現形式としては、一旦認識対象文（タスク）の状態遷移をオートマトンで表現したならば、ユーザが機械的にテキスト形式に変換できる様な表現形式を考える。

2)認識部の内部表現への変換。

ユーザは、記述したテキスト形式のタスクファイルを、ホストコンピュータ上で動作する変換プログラムによって認識部の内部表現形式のタスクファイルに変換する。

2)タスクの選択と認識。

ユーザは目的の内部表現形式タスクファイルを選択し、音声認識装置に指定することにより目的のタスクを実行する。

一般的な記述式を説明する前に、簡単な例について説明する。図３は、簡単なロボット制御文を表現するオートマトン表現である。このオートマトン表現をユーザは図４の形式で入力する。ここで、λはナル遷移を表現するもので、これによってオートマトンの表現がコンパクトになる。しかし、2.2.2で述べたアルゴリズムでナル遷移に対しては特別な処理が必要になる[9]。

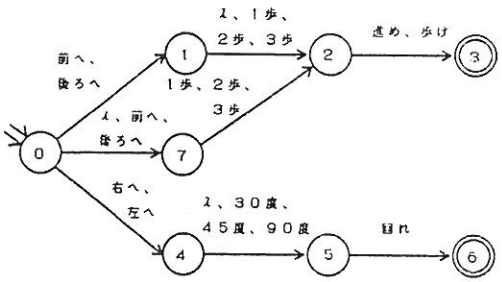

図３　簡単なロボット制御文のオートマトン表現例

```
TITLE  ロボット制御１;
WORD   前へ、後ろへ、右へ、左へ、進め、
       歩け、１歩、２歩、３歩、３０度
       ４５度、９０度、回れ;
LINK   0, 1, 前へ, 後ろへ;
LINK   0, 4, 右へ, 左へ;
LINK   0, 7, １歩, ２歩, ３歩;
LINK   7, 2, 前へ, 後ろへ;
LINK   1, 2, λ, １歩, ２歩, ３歩;
LINK   2, 3, 進め, 歩け;
LINK   4, 5, λ, ３０度, ４５度, ９０度;
LINK   5, 6, 回れ;
MSTA   7;
FSTA   3,6;
FILE   ROBOT1;
END;
```

図４　図３に対応するタスク定義例

例からも推察できるように、ユーザによるオートマトン表現の入力形式は以下のように定義される。

TITLE ;ユーザが分かりやすい名前をつける。
WORD ;認識対象文に現れる全ての単語を、' 'で区切って並べる。ここで'単語'とは、状態遷移を行うときに現れる'単語'と定義し、'単語'単位に音声標準パターンが用意される。
LINK ;第１要素:遷移元の状態番号、
第２要素:遷移先の状態番号、
第３要素以降:その遷移のときに現れる単語。
ただし、'λ'はナル遷移を示す。
MSTA ;オートマトンの状態数を表す。
（ただし、初期状態は除く）
FSTA ;オートマトンの最終状態を表す。
FILE ;対応する標準パターンのファイル名。
END ;記述の終了。
各宣言の終了は';'で示す。

4.2　ユーザ定義による意味情報抽出法

音声認識装置をマン・マシンインターフェースとして実際に応用するためには発声された文を認識する能力にとどまらず認識結果に対して的確に応答する能力が必要とされる。つまり認識結果からマシンの応答に必要なパラメータ（応答パラメータ）を抽出する能力が求められる。ここで認識文から応答パラメータを得るプロセスを意味情報抽出と呼ぶ。

認識結果から応答パラメータを得る際に考慮しなければならない点を列挙する。

①単語の出現順序に依存しない場合には同一のパラメータを返す必要がある（例１）。
②単語の出現順序に依存する場合には出現順序も保存した形でパラメータを返す必要がある（例２）
③一文中に同一の単語が複数回現れる場合には同一単語の構文上の意味も考慮しなければならないが、有限状態オートマトン表現の場合には遷移先の状態番号で区別できるので、記述に際して遷移先の状態番号を入れる（例３）。

例１　・前へ　１歩　進め
　　・１歩　前へ　進め
例２　・右へ　回り　前へ　２歩　進め
　　・前へ　２歩　進み　右へ　回れ
例３　・１１時発のひかり１１号

以上の点を考慮して意味抽出の表現形式として以下のような形式を用いる。

まず例として図４に対応した意味抽出の表現を示す。

```
TASK ZENGO
   {
      {[2,3];進め,歩け},
      {[0,1],[0,7];前へ,後ろへ},
      {[1,2],[7,2];１歩,２歩,３歩}
   }
TASK SAYUU
   {
      {[5,6];回れ},{[0,4];右へ,左へ},
      {[4,5];９０度,３０度,４５度}
   }
```

認識例と、それに対応した戻り値を示す。

１．　右へ　３０度　回れ　：
　ZENGO[0]＝0　SAYUU[0]＝1
　ZENGO[1]＝0　SAYUU[1]＝1
　ZENGO[2]＝0　SAYUU[2]＝2
２．　２歩　進め　：
　ZENGO[0]＝1　SAYUU[0]＝0
　ZENGO[1]＝1　SAYUU[1]＝0
　ZENGO[2]＝2　SAYUU[2]＝0

ZENGO[0]やSAYUU[0]が0の場合はその動作の要素は現れなかったこととする。例２のような入力文に対しては、２つのタスク（ZENGO,SAYUU）の意味表現とタスクの起動順序を抽出しなければならない。このため、タスクの起動順序を別に抽出・保存する。

一般的な表現を以下に示す。

```
TASKタスク名
{(S1,S2,..,Si;T1,T2,..,Tj;
  S1,S2,..,Sk;T1,T2,..,Tl;..;..),
 (S1,S2,..,Sm;W1,W2,..,Wn),
 (..;..),...,(..;..)}
```

タスク名:ユーザ定義のタスク変数名.
Sa,Sb:状態番号.

Si,Sk:タスク決定単語の遷移状態番号対. ;[Sa,Sb]
Tj,Tl:タスク決定単語.
Sm:遷移先状態番号対. ;[Sa,Sb]
Wm:応答パラメータとして値を返す必要のある単語名.(Wlが'default'値となる。但しwlが'_'の場合はdefault値を設定しない。)

5. 装置の評価と高速化

5.1 装置の評価

本音声認識装置の仕様は表1に示したが、本節ではユーザインタフェースを含めた認識システムとしての評価を行う。認識タスクとしては、タスク1(桁数未知の連続数字、語い数 10,状態数 2、平均分岐数 10)とタスク2(ロボット制御命令文、語い数 29,状態数14,平均分岐数 4)の2種類を用意した。認識方式は特定話者方式で標準パターン作成用に各単語を一回発声し登録した。

認識精度はタスク1、タスク2共に文認識率で90%以上を得ている。認識時間はタスク1では実時間の2倍、タスク2では、実時間の10倍程度である。

マッチング処理を行うImPPと後処理を行うホストコンピュータは並列に動作している。マッチング処理の速度と後処理の速度を計測し比較するとタスク1で3: 5、タスク2で1: 2であった。このことからマッチング部を高速化しても後処理がネックになってくる事が分かる。

ImPPの処理効率を示す尺度として稼働率[13]がある。現在のプログラムでの稼働率は主にアドレス発生に用いているImPP1、ImPP3が平均5%、マッチング及び最適パス選択に用いているImPP2、ImPP4が平均24%である。マッチング部の処理速度の向上は、使用ImPP数の増加、特徴パラメータの次元圧縮(例えば8次LPCメルケプストラム)、ベクトル量子化(VQ)[11]等の手法が考えられる。しかしImPPで行っているマッチング処理のうち市街地距離計算に要する処理時間は全体の23%であり、前出の方法により市街地距離計算が軽減されてもマッチング処理部全体としての認識時間の軽減は2割程度にとどまる。よってマッチング部の高速化はImPPのマッチングプログラムを最適化し稼働率の向上図る事や多重並列化が必要である。

一方、複雑なオートマトンで表現されるタスクでは、後処理の方が時間がかかる。よって最終的には後処理を高速化せざるを得ない。後処理の高速化手法については次節で述べる。

ユーザインタフェースは、従来の認識装置と比較してユーザによる認識タスクの指定を非常に容易にした。ユーザは認識装置の起動時にタスクファイルを指定するだけで目的の認識タスクを実行できる。

5.2 後処理の高速化

後処理の高速化手法としてはビームサーチ手法や、単語の平均マッチング距離にしきい値を設ける手法が考えらる。

単語の平均マッチング距離にしきい値を設ける方法は、マッチング部によって計算された各単語の平均マッチング距離をしきい値によって選別し、選別された単語についてのみ後処理を行う。これは、特に語い数が多い時に有効であると考えられ1/5程度の処理の削減が期待できる。

ビームサーチ手法は、後処理軽減のための一手法である。ビームサーチ手法は処理している入力フレーム上において遷移可能性の高い状態番号を上位からR個選択し(Rを通常ビーム径と呼ぶ)その状態に続く遷移単語についてのみ後処理を行う手法である。この手法は特に状態数の多いタスクに対して有効であると考えられ、これも認識精度を落とす事なく1/5程度の処理削減が期待できる[6]。

これらの手法は共に平均分岐数の多い複雑なタスクほど高速化の効果が高い。これらの手法により後処理の高速化を図った場合、認識精度はそのままで10倍以上の後処理の高速化が図れると思われる。それ故、これらの手法を採用することによって連続数字の実時間認識が可能になる。

6. 連続音声認識装置のLOGOへの応用

著者らの開発した連続音声認識装置及びユーザフレンドリなインターフェースのアプリケーションとして、CAI指向のプログラミング言語として知られているLOGOへの適用を行っている。LOGOには、命令文を日本語で表現することの出来る日本語LOGOを用いた。認識タスクに日本語LOGOを採用することにより日本語音声による対話形式のプログラム環境を構築することが出来る(図5)。

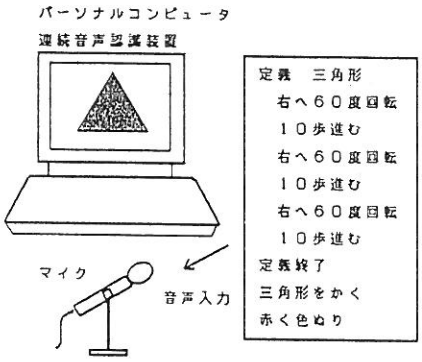

図5 音声入力による日本語LOGOの使用例

今後、このアプリケーションを通して、音声認識装置のユーザにとって好ましい音声入力環境、対話法等の問題について研究する予定である。

認識タスクは、日本語LOGOのプログラミングに必要なタートルグラフィックスを中心とする基本命令語約100単語によって構成されており、平均分岐数約5の有限オートマトン(図6)で表現されている。日本語LOGO文認識タスクを連続音声認識装置に適用するためのタスク表現及び意味抽出表現を図7、図8に示す。

単語を基本ユニットとする連続音声認識装置を日本語LOGOのようなプログラミング言語の音声入力に応用しようとする場合、ユーザ定義の変数やプ

ロシジュアの登録に伴う未知語の入力に対する処理の必要性が生じる。この様な場合、新しい定義語（未知語）は装置では認識出来ないので適当なシンボルを付して入力音声パターンそのものを標準パターンに登録することによって対処する予定である。また未知語の登録と同時に、有限オートマトンによるタスク表現にも必要に応じて自動的に変更、追加等の処理を行わなければならない。一方、認識誤りやユーザの問い合わせ文に対しても対処しなければならない。これらの問題は、プログラミング言語の音声入力問題に限らず音声入力装置をとうしてマン・マシン間のコミュニケーションを行う際に生じる一般的な問題であり今後の研究課題の一つとなるであろう。

現在のところ日本語ＬＯＧＯ文認識タスクの予備認識実験では、男性話者１名が１０３単語を１回発声登録し、３０文を認識したところ文認識率90%を得ている。認識誤りは数値表現を含む文で生じ、例えば、

・入力：右へ　四　十　五　度　回る
　出力：右へ　九　十　五　度　回る
・入力：歩幅は　三　十　二
　出力：歩幅は　三　十　八

一方、正しく認識できた２７入力文は、例えば、

・左へ　百　八　十　度　回る
・青い　三角形　を　かく
・タイトルの　定義終了

等の入力文であった。

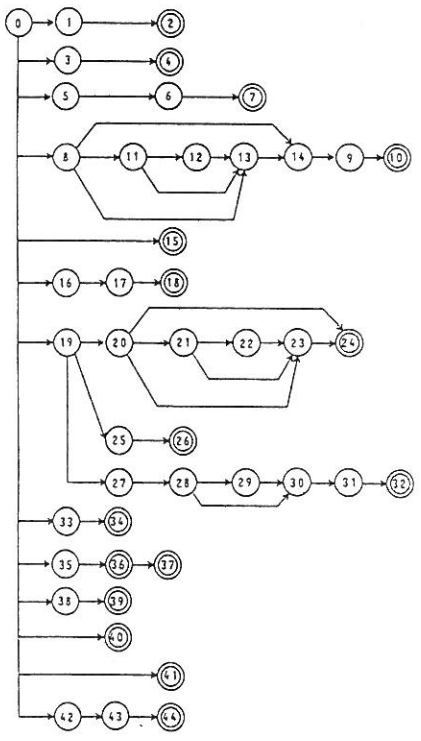

図６　日本語ＬＯＧＯ文の有限オートマトン表現

```
title   日本語ＬＯＧＯサブセット;
word    タートルを, 色の, 歩幅の, 座標の, 角度の,
        出す, 消す, 出力, 設定, 右へ, 左へ, 前へ, 後ろへ, 移動,
        進む, 回転, 回る, 一歩, 二歩, 三歩, 四歩, 五歩, 六歩, 七歩,
        八歩, 九歩, 十歩, 度, もとにもどる, 絶対移動, 色塗り,
        ペンあげ, ペンさげ, 赤い, 青い, 白い, 黒い, 黄色い,
        三角形を, 正方形を, 円を, 長方形を, 置く, かく, A, B, C,
        X, Y, Z, 色, 歩幅, 角度, 一, 二, 三, 四, 五, 六, 七, 八, 九, 十,
        百, は, を, に, イコール, たす, ひく, わる, かける, 数値表示,
        代入, 画面を, センテンスの, タイトルの, ワードの,
        リストの, 変数の, 手続きの, 初期化, 定義, 定義終了, 表示,
        取り消し, ファイルの, プログラムの, 読み込み, 書き込み,
        印刷, 編集, オープン, クローズ, コピー, 削除, 繰り返し,
        やり直し, 使い方, 終了, 実行, はい, いいえ;

link    0, 1, タートルを;
link    1, 2, 出す, 消す;
link    0, 3, 色の, 歩幅の, 座標の, 角度の;
link    3, 4, 出力, 設定;
link    0, 5, 右へ, 左へ, 前へ, 後ろへ;
link    5, 6, _, 一歩, 二歩, 三歩, 四歩, 五歩, 六歩, 七歩, 八歩, 九歩,
              十歩, A, B, C, X, , Z;
link    6, 7, 移動, 進む;
link    0, 8, 右へ, 左へ;
link    8, 9, _, A, B, C, X, Y, Z;
link    9, 10, 回転, 回る;
link    8, 11, _, 百;
link    11, 12, _, 二, 三, 四, 五, 六, 七, 八, 九;
link    8, 13, 百;
link    12, 13, 十;
link    13, 14, _, 一, 二, 三, 四, 五, 六, 七, 八, 九;
link    8, 14, 一, 二, 三, 四, 五, 六, 七, 八, 九;
link    14, 9, 度;
link    0, 15, もとにもどる, 絶対移動, 色塗り, ペンあげ, ペンさげ;
link    0, 16, _, 赤い, 青い, 白い, 黒い, 黄色い;
link    16, 17, 三角形を, 正方形を, 円を, 長方形を;
link    17, 18, 置く, かく;
link    0, 19, A, B, C, X, Y, Z, 歩幅, 角度;
link    19, 20, は, イコール;
link    20, 21, _, 百;
link    21, 22, _, 二, 三, 四, 五, 六, 七, 八, 九;
link    20, 23, 百;
link    22, 23, 十;
link    23, 24, _, 一, 二, 三, 四, 五, 六, 七, 八, 九;
link    20, 24, 一, 二, 三, 四, 五, 六, 七, 八, 九;
link    19, 25, を;
link    25, 26, 数値表示;
link    19, 27, に;
link    27, 28, A, B, C, X, Y, Z;
link    28, 29, たす, ひく, わる, かける;
link    29, 30, A, B, C, X, Y, Z;
link    28, 30, _;
link    30, 31, を;
link    31, 32, 代入;
link    0, 33, 画面を;
link    33, 34, 消す, 初期化;
link    0, 35, センテンスの, タイトルの, ワードの, リストの, 変数の,
              手続きの;
link    35, 36, 定義, 定義終了, 表示, 取り消し;
link    36, 37, _;
link    0, 38, ファイルの, プログラムの;
link    38, 39, 読み込み, 書き込み, 印刷, 編集, オープン, クローズ,
              コピー, 削除, 表示;
link    0, 40, 繰り返し, やり直し, 使い方, 終了, 実行;
link    0, 41, はい, いいえ;
link    0, 42, 色;
link    42, 43, は;
link    43, 44, 赤い, 青い, 白い, 黒い, 黄色い;

ista    0;
fsta    2, 4, 7, 10, 15, 18, 24, 26, 32, 34, 36, 37, 39, 40, 41, 44;
nsta    44;
file    TLOGO1;
end;
```

図７　日本語ＬＯＧＯ文のタスク表現

```
TASK  TURTLE1 {{[ 1,  2];出す,消す)}
TASK  TURTLE2 {{[ 5,  6];一歩,二歩,三歩,四歩,五歩,六歩,七歩,八歩,九歩,
                         A,B,C,X,Y,Z),
               {[ 0,  5];右へ,左へ,前へ,後ろへ)}
TASK  TURTLE3 {{[ 0,  8];右へ,左へ),
               {[ 8, 14];_,一,二,三,四,五,六,七,八,九),
               {[ 8, 13];_,百),
               {[11, 13];_,十),
               {[11, 12];_,二,三,四,五,六,七,八,九),
               {[13, 14];_,一,二,三,四,五,六,七,八,九),
               {[ 8,  9];_,A,B,C,X,Y,Z)}
TASK  ZUKEI   {{[ 0, 16];赤い,青い,白い,黒い,黄色い),
               {[16, 17];三角形を,正方形を,円を,長方形を)}
TASK  VAR     {{[ 3,  4];出力,設定),
               {[ 0,  3];色の,歩幅の,座標の,角度の)}
TASK  COMP    {{{[19, 20];は,イコール;[25,26];数値表示;[31,32];代入),
               {[ 0, 19];A,B,C,X,Y,Z,歩幅,角度),
               {[27, 28];A,B,C,X,Y,Z),
               {[29, 30];A,B,C,X,Y,Z),
               {[28, 29];たす,ひく,かける,わる),
               {[20, 24];_,一,二,三,四,五,六,七,八,九),
               {[20, 23];_,百),
               {[21, 23];_,十),
               {[21, 22];_,二,三,四,五,六,七,八,九),
               {[23, 24];_,一,二,三,四,五,六,七,八,九)}
TASK  DISP    {{[33, 34];消す,出す)}
TASK  DEF     {{[25, 36];定義,定義終了,表示,取り消し),
               {[ 0, 25];センテンスの,タイトルの,ワードの,リストの,
                         変数の,手続きの)}
TASK  EDIT    {{[38, 39];読み込み,印刷,オープン,クローズ,コピー,
                         書き込み,編集,削除,表示),
               {[ 0, 35];プログラムの,ファイルの)}
TASK  SYS     {{[ 0, 40];繰り返し,やり直し,使い方,終了,実行)}
TASK  IRO     {{[43, 44];赤い,青い,白い,黒い,黄色い)}
TASK  RES     {{[ 0, 41];はい,いいえ)}
```

図８　日本語ＬＯＧＯ文の意味抽出表現

7．むすび

マッチング部をマルチプロセッサで、オートマトン制御による単語連結処理をホストコンピュータで実現することにより１４０単語の語い数からなる連続音声認識が可能な認識装置を開発した。また、ユーザがこの認識装置を容易に応用できるように認識対象文と認識結果の意味抽出法がそれぞれユーザによって容易に定義できるインタフェースを作成した。特に後者は従来ほとんど考慮されていなかったことで、音声認識装置の実用化・普及には欠かせない機能だと考えられる。

さらに、ユーザインターフェースを用いて本音声認識装置を日本語ＬＯＧＯに適用した。今後は、本装置及び実際のアプリケーションである日本語ＬＯＧＯをとうして、音声と機械との対話における実際的な問題点を明らかにしていく予定である。

謝辞　本認識装置の開発にあたり、TiP-PC 及びDSP-320の借用を快く許可していただきました日本電気㈱、及び中部電機㈱の関係各位に深謝します。また、拡張連続ＤＰ法のインプリメンテーションには、本学大学院生坂井秀次氏（現在日本IBM㈱）、安倍慈久仁氏（現在インテルジャパン㈱）の協力を得たことを付記する。

参考文献

(1)　中川聖一："拡張連続DP法による連続音声認識アルゴリズム"，信学論誌，Vol.67，10，pp.1242-1249 (1984)．

(2)　H. Sakoe："Two-Level DP-Matching algorithm a dynamic programming based pattern matching algorithm for connected word recognition"，IEEE，Trans. Acoust，Speech，Signal Processing，Vol，ASSP-27，6，pp.588-595(1979)．

(3)　J. Gaurain and J. Mariani: "A method for connected word recognition and word spotting on a microprocessor"，Proc. Int. Conf. Acoust. Speech, Signal Processing，pp.891-894 (1982)．

(4)　S.E. Levinson and K.L. Shipley : "A conversational-mode airline information and reservation system using speech input and output"，Bell System Technical Journal，59，1，pp.119-137 (1980)．

(5)　中田和男："音声情報処理の最近の動向に寄せて"，日本音響学会誌，42，12，pp.920-923(1986)．

(6)　H. Sakoe: "A generalized Two-Level DP-Maching algorithm for continuous speech recognition"，信学論誌，Vol.65-E，11，pp.649-656(1982)．

(7)　S. Nakagawa: "Connected word recognition algorithm by Constant Time Delay DP，O(n)DP and Augmented Continous DP matching"，information Science，Vol.33，pp.63-85(1984)．

(8)　安倍，坂井，中川："拡張連続DP法による連続音声認識装置の試作"，ディジタル信号処理シンポジウム，pp.141-146 (1986-11)．

(9)　中川聖一："縮退化したオートマトン制御による連続音声認識アルゴリズム"，日本音響学会講論集，2-1-7 (1983-10)．

(10) R.I. Pawate, et al.:"Connected Word Recogniser on a Multiprocessor system"，Proc.ICASSP，pp.1151-1154 (1987)．

(11) N.Sugamura，S.Furui:" Isolated word recognition using string of phoneme-like templates (SPLIT)"，J.Acoust.Soc.Jan(E)5，4，pp.243-pp.252(1984)

(12) ＮＥＣ日本電気株式会社：μＰＤ７２８１ユーザーズ・マニュアル

(13) ＮＥＣ日本電気株式会社：μＰＤ７２８１ソフトウェア・パッケージ取扱説明書